

# Knowledge Representation Artifacts for Use in Sensemaking Support Systems

Jean Roy and Alexandre Bergeron Guyard  
Defence Research and Development Canada  
Quebec City, Canada  
Jean.Roy@drdc-rddc.gc.ca  
Alexandre.Bergeron-Guyard@drdc-rddc.gc.ca

**Abstract**—The development of sensemaking support systems requires that one cares about knowledge representation. Motivated by the fact that no single representation method is ideally suited by itself for all tasks, the authors propose a collection of knowledge representation artifacts appropriate for processing in computer-based support systems for situation analysis. The approach described makes it possible to combine the advantages of different representational forms. Each representation paradigm can be matched to an aspect of sensemaking that is a natural fit with this aspect. For example, representing information as propositions is suitable for automated reasoning, while encoding this information using a graph representation enables knowledge discovery through network analytics techniques. The spatial features are a good fit with geospatial reasoning, while situation cases evidently fit well with the case-based reasoning paradigm. These representation artifacts (and a few others) are briefly described in the paper, and some directions for future work are discussed.

**Keywords**—sensemaking; situation analysis; knowledge representation; support system

## I. INTRODUCTION

The notion of awareness has to do with having knowledge, cognizance or understanding [1]. In turn, sensemaking can be seen as the process of creating situation awareness in situations of uncertainty [2, 3]. It is a constant process of acquisition, reflection, and action. It is an action oriented cycle that people continually and fairly automatically go through in order to integrate experiences into their understanding of the world around them [4]. The considerations above suggest the adoption of a knowledge-centric view to situation analysis and sensemaking support systems [5-6]. Such a view ultimately requires that one cares about knowledge representation, a discipline concerned with how knowledge can be represented symbolically and manipulated (processed and/or communicated) in an automated way by computer programs, in particular those simulating human reasoning.

From a related perspective, in contemporary activities, analysts and decision-makers at all levels works in an information-saturated environment. The staffs need enough information to make decisions, but also need to be supported by technology so that they are not overwhelmed with information. Unfortunately however, although significant progress has been achieved in recent years, the processing of a large proportion of the data and information made available from the ever increasing number and variety of sources is still being performed manually. Of course, manually and mentally

processing huge amounts of data and information is very laborious, complex, time consuming and prone to error. Actually, the amount and complexity of data and information now available have made this type of processing impractical and the situation is worsening as more and more data and information sources are developed and become available. Mental and manual processing must be replaced by automated processing wherever it makes sense and is possible.

Clearly, given the data and cognitive overload issues mentioned above, automation has a critical role to play in sensemaking and decision processes. This, coupled with the adoption of a knowledge-centric view to situation analysis and decision-making as previously discussed, has lead to the development of several automated processing components for use in sensemaking support systems [6-11]. In turn, automated processing has required the development of appropriate knowledge representation mechanisms to communicate situation knowledge to the computer-based processing components, and to collect the results of the processes.

Aligned on these lines of thought, this paper describes a set of formal knowledge representation artifacts that have been developed in order to represent knowledge in a formal way suitable for processing in computer systems. These artifacts have been conceived to meet the needs of the Sensemaking Support System ( $S^3$ ) developed at Defence Research and Development Canada (DRDC). The  $S^3$  is a federation of innovative, computer-based, composable and interoperable sensemaking support tools, which are integrated and interleaved into an overall, continuous process flow supporting the analysts involved in situation analysis activities.

The paper is divided as follows. Knowledge-based systems are briefly introduced in Section II, while the knowledge representation artifacts developed for the  $S^3$  are globally presented in Section III. Then, Sections IV to X succinctly describe each of these artifacts. Section XI discusses additional artifacts that are used to represent the know-how of domain experts and, finally, some concluding remarks and themes for future work are presented in Section XII.

## II. KNOWLEDGE-BASED SYSTEMS

Given the intrinsic nature of sensemaking and situation awareness, most of the components of the  $S^3$  involve knowledge-based system (KBS) and semantic Web technologies. A KBS is a computer system that represents and uses knowledge to carry out a task. An expert system is an intelligent computer program that uses knowledge and

inference procedures to solve problems. As the applications for the technology have broadened, the more general term knowledge-based system has become preferred by some people over expert system because it focuses attention on the knowledge that the systems carry, rather than on the question of whether or not such knowledge constitutes expertise. For a large portion, the processing components of the  $S^3$  have been built on KBS technologies. The selection of these technologies has been motivated by a number of their intrinsic characteristics, the main one being that processing is separated from the problem-solving knowledge in knowledge-based systems. This characteristic allows:

- to represent knowledge in a more natural fashion,
- the focus to be on capturing and organizing problem-solving knowledge,
- changes to be made to the knowledge base without side effects on program code,
- the same control and interface software to be used in a variety of systems, in different domains, and,
- to experiment with alternative control software for the same knowledge base.

As a result of the attributes mentioned above, the processing components of a KBS are typically:

- generic (i.e., the processing is intrinsically « agnostic »; it's the a priori knowledge of a particular domain that makes the processing specific),
- developed “only once” (or more precisely, the exact same components can be used/reused in different application domains without any modifications being required), and,
- developed by “others” (i.e., they are developed, tested, debugged, etc. by others and then made available from open sources or commercially).

In the context of the  $S^3$ , the use of KBS technologies allows for the scientists at DRDC to first develop a single, unique system, and then to exploit it under different research projects, for various customers in diverse domains.

### III. KNOWLEDGE REPRESENTATION ARTIFACTS FOR SENSEMAKING SUPPORT SYSTEMS

The object of knowledge representation (KR) is to express knowledge in computer-tractable form, such that it can be exploited. KR and reasoning is the area of artificial intelligence (AI) concerned with how knowledge can be represented symbolically and manipulated in an automated way by reasoning programs. KR research studies the problem of finding a language in which to encode the knowledge so that the machine can use it. It should support the tasks of acquiring and retrieving knowledge, as well as subsequent reasoning.

One may be under the impression that a knowledge engineer must find a single best knowledge representation and stick with it. However, it is not necessary to select and use only one representation paradigm in a KBS. Actually, no single

knowledge representation method is ideally suited by itself for all tasks. An important alternative is the use of multiple representations, which makes it possible to combine the advantages of different representational forms. From a related perspective, when using several sources of knowledge simultaneously, the goal of uniformity may have to be sacrificed in favour of exploiting the benefits of multiple knowledge representations, each tailored to a different subtask. In view of the discussion above, an approach with multiple representation paradigms has been retained for the  $S^3$ .

A variety of knowledge representation paradigms, schemes and techniques have been devised in the AI community over the years. These includes lists and outlines, decision tables, decision trees, state and problem spaces, production rules, subject-predicate-object triples, semantic networks, schemata, frames, scripts, logics, ontologies, etc. Fig. 1 depicts the specific knowledge representation artifacts that have been specified and developed at DRDC for use in sensemaking support systems.

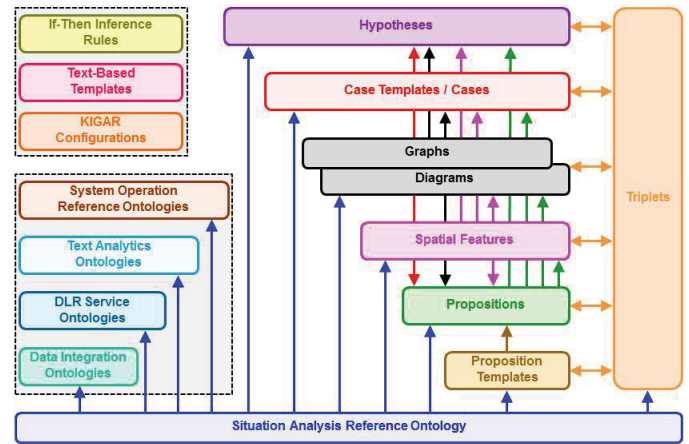


Fig. 1. Knowledge representation artifacts for sensemaking support systems

In multiple representations, more than one symbol structures are used to designate a thing in the environment. The necessity of translating among knowledge representations thus becomes an issue in these cases. Moreover, measures must be taken to keep the representations synchronized whenever multiple representations are used. This issue is inherent in the use of multiple representations. Nevertheless, the  $S^3$  uses multiple knowledge representation schemes, which are described at a high level in the remaining of this paper.

### IV. ONTOLOGIES

Many definitions of the term *ontology* have been proposed by a variety of authors. Among these, the definition proposed by [12] based on the work of [13] seems appropriate for the development of knowledge-based situation analysis support systems: “An ontology is a formal, explicit specification of a shared conceptualization.” During the last decade, increasing attention has been focused on ontologies and ontological engineering. Ontologies are now widely used in knowledge engineering, artificial intelligence, computer science, knowledge management, natural language processing, and many other fields.

### A. Situation Analysis Reference Ontology

Within our knowledge representation framework, the high-level objective for the Situation Analysis Reference Ontology (SARO) is to provide a shared, collective semantic resource that constitutes the semantic foundation for the overall set of representation artifacts. The approach pursued is to develop the SARO as an evolving collection of shared plug-and-play ontology modules. Fig. 2 provides a snapshot of a subset of the SARO, here shown in the Protégé tool.

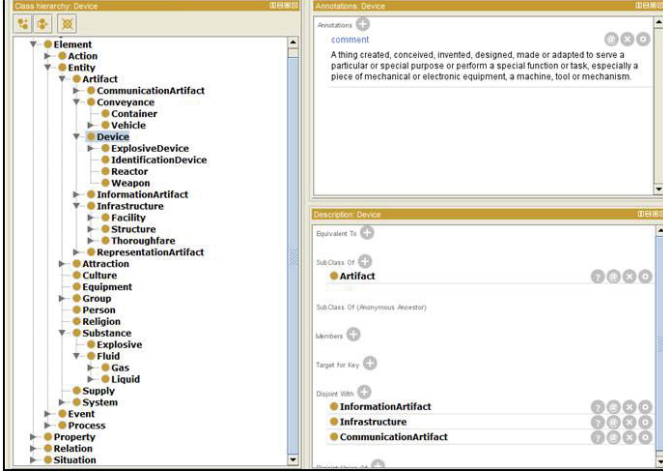


Fig. 2. Situation analysis reference ontology shown in Protégé

### B. Other Ontologies

The SARO enables the subsequent development of a variety of application ontologies including, for example, ontologies exploited for the automated annotation of unstructured text documents in text analytics applications, the integration of data from multiple, heterogeneous sources in support of concepts such as that of a Unified Data Space (UDS), automated reasoning in applications that use ontologies to encode the problem-solving know-how knowledge of Subject Matter Experts (SMEs), and knowledge representation in network analytics applications such as those for Social Network Analysis (SNA). The development of any such application ontology would extend the SARO to achieve a specific purpose, while enforcing the principles and rules initially applied to the SARO.

## V. PROPOSITION TEMPLATES AND PROPOSITIONS

A proposition is an expression in language or signs (e.g., a particular word, phrase, or form of words), a statement (a single declaration, sentence, assertion or remark; a message that is stated or declared; a communication (oral or written) setting forth particulars or facts, etc.), that affirms or denies something, which can be believed, doubted or denied, and that can be significantly characterized as either true or false. Some examples of propositions are “John is a person”, “Ship X is conducting activity Y”, and “Passenger list of flight X includes person name Y”. And there is obviously an infinite number of such things that can be affirmed or denied. It is worth noting that propositions can be very simple (e.g., “John is a person”, or “John has a weapon”) or much more elaborated (e.g., “John has rendezvous with Mike in the park at 20:30 to discuss the

plan”). Fig. 3 illustrates a proposition formulated in both a natural language and a formal language.

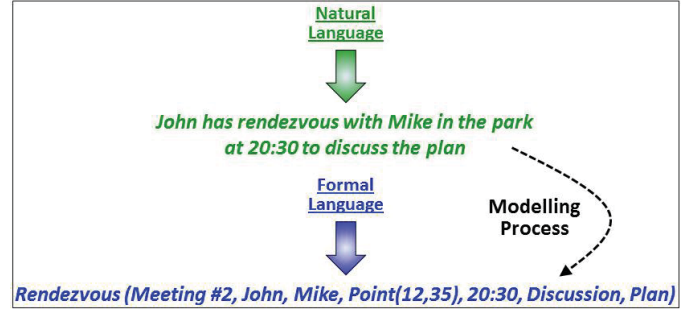


Fig. 3. A proposition in natural and formal languages

In general, propositions are formulated in natural languages (English, French, German, etc.), for communication and processing purpose by human beings. Exploiting the notion of a proposition in a computer system however requires the expression of propositions through formal languages, i.e., in a format that is more suitable for communication and processing by computer systems.

In our framework, the three main components of a formal proposition are: 1) the main statement, 2) the triplet mappings, and 3) the attributes. The main statement is a mandatory component of the formal proposition. It represents the essence, the core of what is expressed with the proposition. Optional triplet mappings can be attached to the main statement in order to provide all sorts of amplification data, typically regarding the arguments used for the statement. Finally, all sorts of attributes (predefined or not) can also be attached to the proposition in order to further qualify it.

A distinction is established between a “proposition template” and a “proposition instance”. As shown with Fig. 4, the idea is that one or numerous propositions can be created (or instantiated) from a single proposition template.

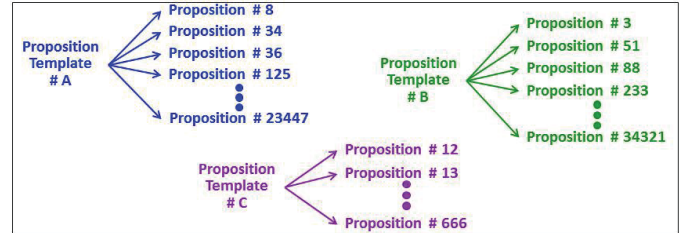


Fig. 4. Multiples proposition instances from a single proposition template

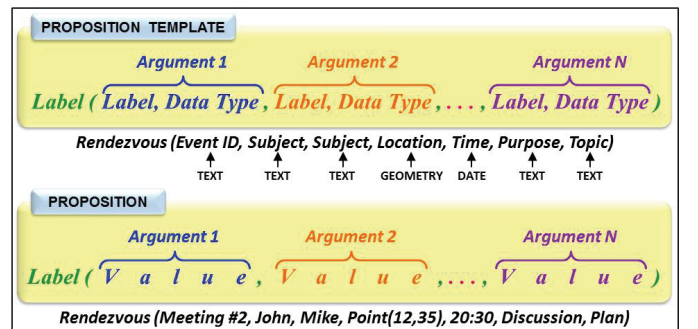


Fig. 5. Structure of a proposition template and corresponding proposition



A proposition template must first be created to specify the base structure of propositions to be represented, and one or multiple propositions can then be instantiated following the model established by the template. As shown on Fig. 5, a proposition template is defined by a label and a list (of a variable length) of arguments of a precise type, listed in a precise order. The argument label is used to distinguish the different pieces required for the proposition, and the argument type is used to restrict the values that can be set for this argument.

In our framework, a triplet mapping is a combination of “*Subject–Predicate–Object*” that is used to assert a correspondence between items that play a role in the context of a proposition. Such triplet mappings can be attached (optionally) to the main statement of a proposition in order to provide all sorts of amplification data, typically regarding the arguments used for the statement. Fig. 6 illustrates the concept.

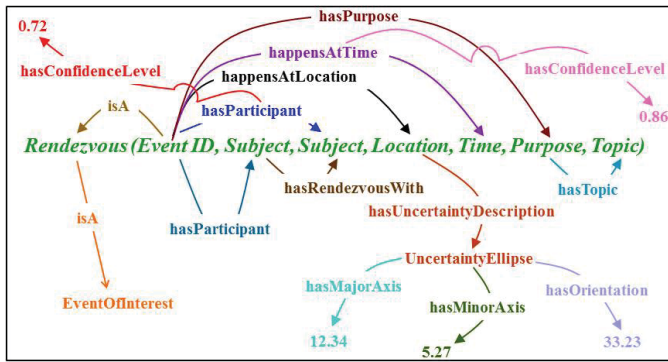


Fig. 6. Triplet mappings proving additional details about the proposition

Fig. 7 is another example of using triplet mappings, showing in particular a “Predicate on Predicate” mechanism.

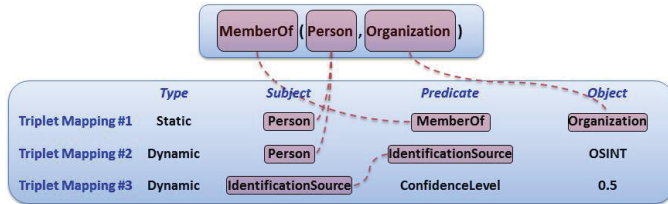


Fig. 7. A simple example of the “predicate on predicate” mechanism

As shown in Fig. 7, each triplet mapping has a type that specifies if it is static or dynamic. Static triplet mappings have a fix value provided by the knowledge engineer during the definition of the proposition template. Dynamic triplet mappings have an initial default value provided by the knowledge engineer during the definition of the proposition template, but this value can be changed later during the exploitation of the individual propositions built on the corresponding proposition template.

Finally, a number of attributes can also be attached to a proposition (as metadata) to support the manipulation and management of the proposition, and to provide a wide variety of contextual information for the proposition regarding uncertainty, temporal and spatial issues, security, the source of the proposition, etc.

## VI. SPATIAL FEATURES

A *spatial feature* knowledge representation artifact is a geo-located geometric shape used to represent (i.e., to model) a real world object or a concept (e.g., a point of rendezvous, a border between two countries, an interdiction zone, a moving storm, etc.) that is deemed useful for the analysis of a situation. A spatial feature is:

- A geometric shape: A spatial feature is a point, a line (straight line, waypoint line), or a surface (circle, rectangle, ellipse, polygon, etc.).
- A geographically located geometric shape: At a given moment in time, the spatial feature is located at a precise position in the world.
- A geographically oriented geometric shape: At a given moment in time, the spatial feature is oriented in a precise direction in the world.
- An evolving geometric shape: It is possible for the geometry of a spatio-temporal spatial feature to change as time advances.
- A moving geometric shape: Spatial features can be fix, or moving through the concept of a “motion trajectory” (attached to the spatial feature) that describes the motion in time of the “point of origin” of the spatial feature.
- A semantic geometric shape: Using ontologies, the domain specification of a spatial feature is fully customizable and controlled by the intelligence analysts. A spatial feature can either be domain agnostic (e.g., a point of rendezvous, a zone of exclusion) or domain specific (e.g., a zone closed to fishing, the 200 nm limit at sea around a country).

Fig. 8 illustrates a few examples of spatial features represented on a map.

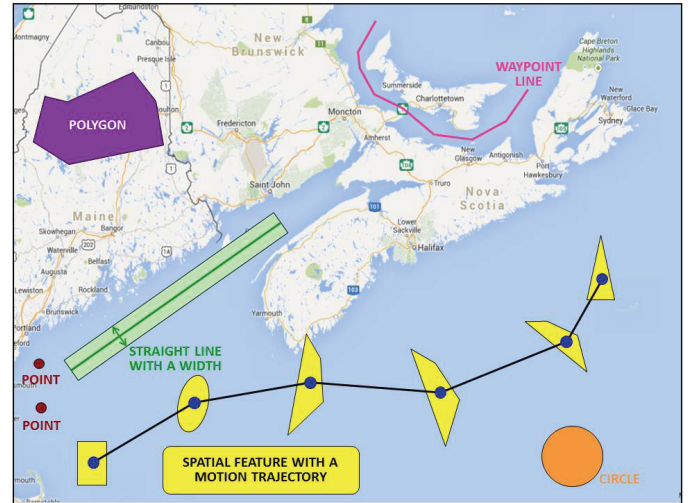


Fig. 8. Examples of spatial features on a map

The main purpose of the spatial feature knowledge representation artifact is to support geospatial analysis with the Kinematic and Geospatial Analysis Reasoner (KiGAR)

automated reasoning service (cf. subsection XI) of the sensemaking support system.

## VII. DIAGRAMS

A graphical language has been defined for the construction of explicit representations (or models) of situations. This language, shown in Fig. 9, allows for the situation modeler to define and manipulate situation model components (SMCs) to create graphical representations of situations. The language is limited to five types of SMCs that can be used by the modeler: 1) Diagram Node, 2) Undirected Diagram Relation, 3) Directed Diagram Relation, 4) Relation Origin Connecting Point, and 5) Relation Destination Connecting Point. Everything that a modeler has to say about a given situation must be expressed using only these five types of SMCs. One should note that only one SMC is required to define a diagram node, while three SMCs are required to define a relationship (i.e., the relation itself, its origin, and its destination).

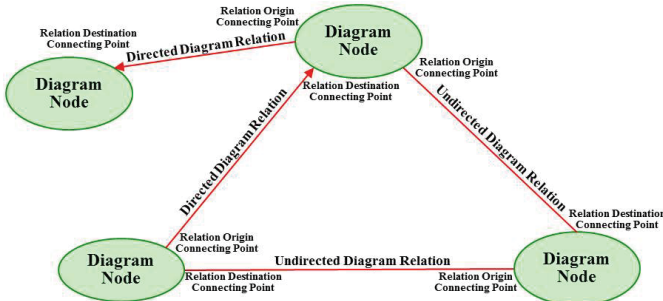


Fig. 9. Graphical language for situation model diagrams

A very important aspect is that the situation model components of types Diagram Node and Relation in Fig. 9 are only « place holders » or « containers ». As such, they don't by themselves convey any particular semantics related to the situation being modeled. It is the actual contents of the situation model components that should make sense (or not) with respect to the situation of interest. There is certainly a precise "container" semantics related to the graphical language itself. For example, the support system understands the meaning of what the "origin of a relation" is from a graph point of view, and what it is allowed (or not) to do with this component from a "container management" perspective, but this semantics is not related to the situation being modeled. This is an important aspect, as the support system can be used in different domains that make sense to the user but that are totally irrelevant for the system itself. One can thus use the support system to describe a "guest and cooking situation", a "maritime drug smuggling situation", an "improvised explosive device situation", etc.

## VIII. GRAPHS

The *Graph* model is another knowledge representation artifact composed of nodes and links. In this case, it has been specifically developed to perform network analyses. Initially created in the context of social network analysis (SNA), it can be applied to any analysis requiring network measures and metrics. As depicted in Fig. 10, the graph is based on three main constituents: "Entity", "Relation" and "Property". An

"Entity" will always be a node and a "Relation" will always be a link between two nodes. As also exposed in Fig. 10, a "Property" can qualify an "Entity" like the *first\_name* of a *person*; it can also qualify a "Relation" like the *start\_date* when a *person* was a *member\_of* a specific *organization* or even a "Property" such as, for instance, the *date* when the *employees\_number* was estimated.

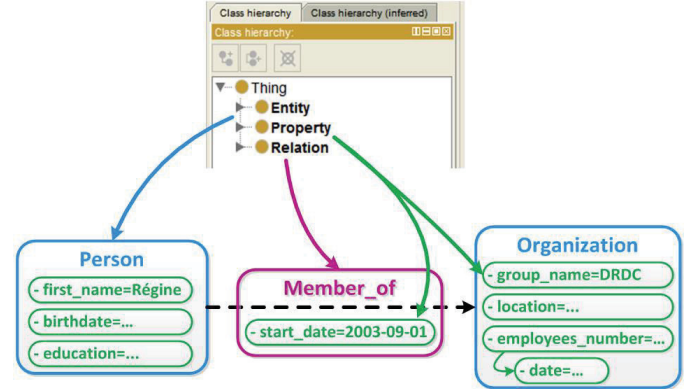


Fig. 10. Knowledge representation artifact for a graph

It is worth noting that a "Property" can qualify other properties with an infinite depth. In our example, the *date*, which is a "Property" could also be qualified through another "Property" such as, for instance, the *uncertainty* about the *date* itself when the *employees\_number* was estimated. This is a unique characteristic of the *Graph* model since current graph standards like GraphML do not yet permit the qualification of a property by another property.

Another particularity of the *Graph* model, aligned with the ontological approach exposed in Section IV, is its foundation based on a network ontology. If network analyses require to be executed on a network of social nature, the entities, relations and properties will be accordingly defined in the ontology and they will be different from the ones detailed for cyber networks. The *Graph* model relies directly on the network ontology that is selected by the end-user and that feeds it automatically and directly.

## IX. CASE TEMPLATES AND CASES

A case-based reasoner solves current problems by using or adapting prior solutions to previous problems. The general idea is to emulate the human reasoning process that relies on past experiences to solve new problems, reusing past solutions. The premise is that new cases will bear sufficient similarity to past problems to allow for an appropriate mapping. In order for it to work, a Case-Based Reasoning (CBR) system requires cases that are stored in a case-base. Typically, a case is composed of a representation of a problem and its solution. A CBR system will attempt to map the new problem to an existing case and its corresponding solution.

Fig. 11 shows the main knowledge representation artifacts that have been developed at DRDC to enable CBR. The *Case Template* is composed of a *Description Template*, a *Conclusion Template*, and *Similarity Measures*.



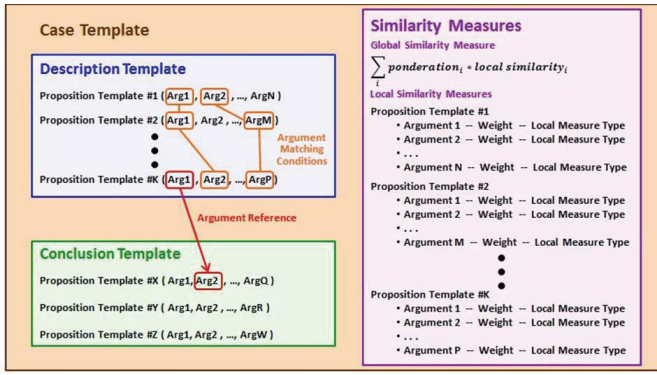


Fig. 11. Knowledge representation artifacts for case-based reasoning

A *Description Template* is built using two essential ingredients: A set of *Proposition Templates* (cf. subsection V) and a set of *Argument Matching Conditions* (AMC). An AMC is the glue that logically links the various proposition templates together. It specifies the arguments that must be similar in order for the proposition templates to form a description template, which can have zero or multiple AMCs.

Using these AMCs, the system will go over all propositions available for the current situation. If propositions matching the proposition templates specified in the description template are found, along with their arguments matching the AMCs, a description will be created (according to the description template) and used to compare against other descriptions present in the case-base. For certain argument types, the user can specify the type of matching condition to be used:

- **Ontology:** The user can specify whether the description's instance must be exactly the same, or of the same class.
- **Distance, Number, Double:** The user can specify whether it is equal, greater or less.
- **Date, Date Interval:** The user can specify if it's the exact date, before, after, within a particular time buffer.
- **Geometry:** The user can specify if the geometry is identical, within, overlapping, close to (within a specified threshold).

Using a particular description template on a given situation, it is possible that more than one description will be populated.

A *Conclusion Template* is built using one or many proposition templates. It can also contain one or many argument references, which link the arguments of the proposition templates contained in the description template to those of the conclusion template. However, it is not mandatory that the proposition templates in the conclusion reuse any of the arguments mentioned in the description template.

Finally, *Similarity Measures* are used to evaluate the similarity between the current situation's description and the descriptions contained in the case-base. They are composed of a global similarity measure and a set of local similarity measures. The former is used to combine the results from the local similarity measures into a single measure. The local similarity measures are used to evaluate the similarity between

the arguments of propositions. A local similarity measure exists for each argument type, and there are eight different types of arguments that can be exploited for propositions.

## X. HYPOTHESES

Uncertainty makes the analysis of even simple situations difficult. It forces analysts to formulate and manage hypotheses during the construction of explicit representations of real world situations. In our framework, there is uncertainty when there are more than one mutually exclusive possibilities for the existence and/or the contents of any given situation model item. This is illustrated in Fig. 12.

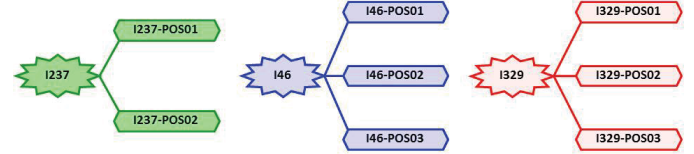


Fig. 12. Items and uncertainty about each individual item

A hypothesis tree data structure is used to keep track of this uncertainty and of the corresponding multiple situation models that must be maintained in parallel.

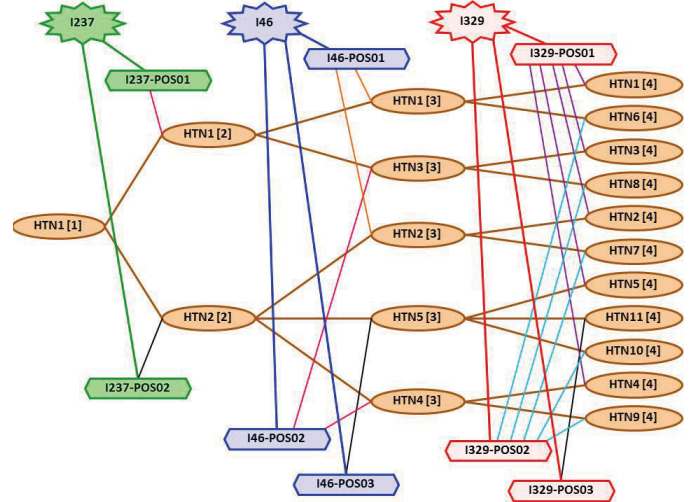


Fig. 13. Hypotheses to represent relationships among item possibilities

Because of human cognitive limitations, formulating and managing hypotheses during the construction of explicit representations of real world situations may quickly become overwhelming, even for the most experienced and capable analysts. In this regard, the data structure shown in Fig. 13 enables the development of multiple hypothesis situation analysis support systems to provide better support to the staffs having to deal with uncertainty in situation analysis

## XI. DOMAIN EXPERT KNOW-HOW

The knowledge representation artifacts discussed above have been developed in order to represent aspects of a situation that one wants to model. However, knowledge-based sensemaking support systems also typically require to represent the knowledge of domain experts. Expertise is a specialized type of knowledge that is known only to a few. It is the

extensive, task-specific and implicit knowledge of the expert that is acquired from training, reading, and experience, and that must be extracted and made explicit so it can be encoded and exploited in support systems.

In our sensemaking knowledge representation framework, a number of artifacts are devoted to the encoding of expert know-how. In particular, “*If-Then*” inference rules can be used with propositions by some automated reasoning engine to infer new propositions. An inference rule defines which pattern of propositions will generate new propositions. Such a pattern of propositions can be seen as domain specific knowledge (typically obtain from a domain expert) specifying which propositions should be deduced based on the existence of other propositions.

Text-based templates represent another type of know-how representation artifact in our framework. They are used by the text processing module of the support system to find precise series of words in unstructured text documents and to extract specific propositions from them. They are composed of text-based elements that needs to be matched against text contents, and a list of conclusions defining which proposition(s) to generate when the pattern is matched.

Finally, our sensemaking support system includes the Kinematics and Geospatial Analysis Reasoner (KiGAR) service, which potentially infers new propositions from the current propositions through automated reasoning based on a collection of kinematics and geospatial analyses. A set of propositions are provided to the service, along with some configuration parameters having values fine-tuned based on the expertise of some domain experts. Then, in an attempt to deduce new propositions, the KiGAR service performs some kinematics and geospatial reasoning on these inputs through the exploitation of analyses selected by the user among a set of distinct “position-related” and “motion-related” analyses that also takes into account user-defined spatial features (cf. subsection VI) in the environment. Each KiGAR configuration constitutes some expert know-how that is encoded within a specialized data structure.

## XII. CONCLUSION

In a context of data and cognitive overload coupled with thoughts on a knowledge-centric view to situation analysis and sensemaking support systems, this paper described a set of formal knowledge representation artifacts that have been developed in order to represent situation knowledge in a formal way suitable for processing in computer systems. These artifacts are used to communicate such knowledge to information processing components, and to collect the results of the sensemaking support processes. The work that was presented here is a significant contribution as the resulting artifacts are formal enough to be exploited in computer systems (as required), while being suitable for easy understanding and manipulation by human analysts or decision makers. It is also worth noting that some of the formal models are currently being considered as appropriate solutions to address the challenging issues related to achieving interoperability at the

knowledge level between computer systems running software components in support to situation analysis and decision making. Such communication at the knowledge level is essential if leveraging is to be achieved between computer-based processing systems deployed in different partner organizations. The work reported here will lead to the development of better, more adequate and interoperable computer-based support systems to best serve the analyst and decision-maker communities. Future work would include the development of the notion of a “knowledge cartridge”, which would further enable the exploitation of a single, unique system under different research projects, for various customers in diverse operational domains.

## REFERENCES

- [1] Bossé, É., Roy, J. and Wark, S., Concepts, Models and Tools for Information Fusion, Artech House, Norwood, MA, 2007.
- [2] Leedom, D.K., “Sensemaking Symposium Final Report”, Vienna, VA: EBR, Inc. October 2001.
- [3] Leedom, D.K., “Sensemaking Experts Panel Meeting Final Report”, Vienna, VA: EBR, Inc. June 2002.
- [4] Kolko, J., “Sensemaking and Framing: A Theoretical Reflection on Perspective in Design Synthesis”, in the 2010 Design Research Society conference proceedings.  
(<http://www.jonkolko.com/writingSensemaking.php>)
- [5] Roy, J., “Combining Elements of Information Fusion and Knowledge-Based Systems to Support Situation Analysis”, Proceedings of Multisensor, Multisource Information Fusion: Architectures, Algorithms, and Applications 2006, SPIE Symposium on Defense and Security, Orlando, Florida, 17-21 April 2006.
- [6] Roy, J. and Bergeron Guyard, A., “A Knowledge-Based System for Multiple Hypothesis Sensemaking Support”, 14th International Conference on Information Fusion (Fusion 2011), Chicago, Illinois, USA, 5-8 July 2011.
- [7] Roy, J. and Bergeron Guyard, A., “Supporting Threat Analysis Through Description Logic Reasoning”, 2012 IEEE Conference on Cognitive Methods in Situation Awareness and Decision Support (CogSIMA), Westin New Orleans Canal Place, New Orleans, LA, USA, March 6-8, 2012.
- [8] Roy, J., “Rule-Based Expert System for Maritime Anomaly Detection”, Proceedings of Sensors, and Command, Control, Communications, and Intelligence (C3I) Technologies for Homeland Security and Homeland Defense VIX, SPIE Defense, Security, and Sensing 2010, Orlando, FL, USA, 5 - 9 April 2010.
- [9] Roy, J. and Davenport, M., “Exploitation of Maritime Domain Ontologies for Anomaly Detection and Threat Analysis”, 2nd International Conference on Waterside Security (WSS 2010), Marina di Carrara, Italy, 3-5 November 2010.
- [10] Roy, J., “Automated Reasoning for Maritime Anomaly Detection”, Proceedings of the NATO Workshop on Data Fusion and Anomaly Detection for Maritime Situational Awareness (MSA 2009), NATO Undersea Research Centre (NURC), La Spezia, Italy, 15-17 September 2009.
- [11] Bergeron Guyard, A. and Roy, J., “Towards Case-Based Reasoning for Maritime Anomaly Detection: A Positioning Paper”, Proceeding of The IASTED International Conference on Intelligent Systems and Control, Cambridge, Massachusetts, USA, 4-6 November, 2009.
- [12] Struder R., Benjamins, V. R. and Fensel, D., “Knowledge Engineering: Principles and Methods”, IEEE Transactions on Data and Knowledge Engineering 25(1-2): 161-197, 1998.
- [13] Gruber, T. R., “A Translation Approach to Portable Ontology Specification”, Knowledge Acquisition 5(2): 199-220, 1993.